

68fil/v2

Manual del Software



CONTENIDO

Sección 1 - Mandos del monitor

A	Ayuda
B	Buscar
C	Cargar
D	Duplicar
E	Ejecutar
I	Inicializar
L	Listar programa
M	Modificar
P	Punto de ruptura
R	Registros
S	Traza sin saltos
T	Traza
V	Volcar memoria
X	Reiniciar el monitor

Sección 2 - Formatos

Cadena	Cadena de caracteres
Número	Números enteros
S28	Formato de comunicaciones

Sección 3 - SERMONES (SERvicios del MONitor)

Fprog	Marca el final de un programa
Getch	Toma un carácter del canal serie
Putch	Pone un carácter en el canal serie
Gets	Toma una cadena de caracteres del canal serie
Puts	Pone una cadena de caracteres en el canal serie
Toupper	Convierte minúsculas en mayúsculas en una cadena
Atoi	Convierte una cadena en un número entero
Itoa	Convierte un número entero en una cadena
Strcpy	Copia una cadena en otra
Strlen	Mide la longitud de una cadena en bytes
Memcpy	Copia memoria
Lnputs	Pone en el canal serie un paso de línea y una cadena
Kbhit	Consulta si se ha pulsado una tecla

Sección 1: Mandos del Monitor

68fil v2 Manual del Software

NOMBRE

A - Ayuda

FORMATO

A

DESCRIPCION

Presenta un listado de los mandos con su formato y una breve aclaración sobre sus parámetros.

EJEMPLO

```

68fil: A

Monitor 68fil (version: 1.2)

=====

Compatible con entrenadores 68fil/v2

Copyright FiloSoft 1989

----- instrucciones -----|----- parametros -----
A          - ayuda           | r = identificador de registro
C          - cargar programa  | v = valor
R [r v]    - registros       | d = direccion de memoria
V [i [f]]  - volcar memoria  | i = direccion inicial de bloque
M d ["c"]  - modificar memoria | f = direccion final de bloque
B i f ["c"] - buscar en memoria | b = byte
D i f d    - duplicar memoria | l = lista de bytes
I i f b    - inicializar memoria | n = numero de punto de ruptura (1..7)
P [n [d]]  - punto de ruptura | c = cadena de caracteres
E [d]      - ejecutar        |
T [d]      - trazar          | [z] "z" parametro opcional
S [d]      - trazar sin saltos |
L [i [f]]  - listar programa  |
X          - reiniciar el monitor |

68fil:

```

NOMBRE

B - Buscar en memoria

FORMATO

B i f ["c"]

donde i = dirección inicial de bloque
f = dirección final de bloque
c = cadena de caracteres (opcional)

DESCRIPCION

Busca una secuencia de bytes entre dos direcciones de memoria. La secuencia puede indicarse mediante una cadena de caracteres. En este caso, ésta se busca en la zona de memoria indicada y se muestra cada vez que es encontrada. En el caso de no incluir una cadena de caracteres en los parámetros, el monitor solicita una lista de bytes a buscar. La lista de bytes se forma con números cuyo valor está entre 0 y FF siguiendo las convenciones normales de escritura de números (ver Números en la sección 2: Formatos). Los bytes que forman la lista se separan entre sí mediante blancos.

EJEMPLOS

Buscar el texto "FiloSoft" entre las direcciones 1000₁₆ y 2FFF₁₆

```
68fil: B 1000 2FFF "FiloSoft"  
<00138A> 4669 6C6F 536F 6674 0D0A 0000 0000 FFFF Filosoft.....  
<002FFA> 4669 6C6F 536F 6674 2031 3938 3900 0000 Filosoft 1989...  
68fil:
```

Buscar una lista de bytes entre las direcciones 1000₁₆ y 2FFF₁₆

```
68fil: B 1000 2FFF  
Introduzca lista de bytes: 46 69 6C 6F 53 6F 66 74  
<00138A> 4669 6C6F 536F 6674 0D0A 0000 0000 FFFF Filosoft.....  
<002FFA> 4669 6C6F 536F 6674 2031 3938 3900 0000 Filosoft 1989...  
68fil:
```

NOMBRE

C - Cargar código S28

FORMATO

C

DESCRIPCION

Pone al sistema en estado de recepción de código S28 (ver S28 en la sección "Formatos"). Una vez lanzado este mando el sistema queda a la espera del código; esto permite que se den instrucciones al sistema que debe enviarlo. Este mando sólo podrá ser utilizado cuando el entrenador 68fil se halle conectado a un ordenador con emulador de terminal (ver manual de instalación). Al finalizar la recepción el monitor vuelve a estar listo para atender mandos, por lo que presenta el mensaje '68fil:'. En caso de que se encuentre algún problema o fallo en la transmisión aparece el mensaje correspondiente y ,esta finaliza.

EJEMPLO

```
68fil: C
Listo para recepción de código S28
68fil:
```

NOMBRE

D - Duplicar

FORMATO

D i f d

donde: i = dirección inicial de bloque

f = dirección final de bloque

d = dirección inicial de memoria destino

DESCRIPCION

Permite copiar el bloque de memoria entre las direcciones i y f en un área que comienza en la dirección d. El bloque de memoria resultante a partir de la tercera dirección es siempre igual al contenido inicial entre las dos primeras direcciones, por tanto cuando el bloque destino se solape con el bloque origen éste se ve alterado.

EJEMPLOS

Situar una copia del contenido de la memoria entre 1000_{16} y $10FF_{16}$ a partir de 1100_{16}

68fil: D 1000 10FF 1100 68fil:

NOMBRE

E - Ejecutar programa

FORMATO

E [d]

donde: d = dirección de memoria (opcional)

DESCRIPCION

Lanza la ejecución de un programa a partir de una dirección que puede ser explícita (parámetro) o implícita (encadenando con ejecución anterior). En caso de no corresponder la dirección a una zona de memoria con programa se producirán errores de ejecución que abortarán la misma presentando el correspondiente mensaje.

Al ejecutarse un programa se activan todos los puntos de ruptura (ver P en esta misma sección) que se encuentren establecidos, por lo que el programa se parará solo al llegar a uno de ellos, o si se alcanza la TRAP de terminación (ver introducción de la sección 3).

EJEMPLOS

Ejecutar un programa con comienzo en la dirección 2000_{16} .

```
68fil: E 2000
Programa terminado normalmente.
68fil:
```

Ejecutar un programa con comienzo inmediatamente detrás del final del anterior programa.

```
68fil: E
Programa terminado normalmente.
68fil:
```

NOMBRE

I - Inicializar memoria

FORMATO

I i f b

donde: i = dirección inicial de bloque
f = dirección final de bloque
b = byte

DESCRIPCION

Permite llenar la memoria entre las direcciones i y f con el byte especificado en b. Es normalmente utilizado para inicializar a cero zonas de variables, o a blancos zonas para textos, etc.

EJEMPLOS

Llenar la memoria entre 1000_{16} y $12FF_{16}$ de blancos (20_{16})

```
68fil: I 1000 12FF 20
68fil:
```

NOMBRE

L - Listar programa

FORMATO

L [i [f]]

donde: i = dirección inicial del listado
f = dirección final del listado

DESCRIPCION

Presenta en la pantalla un listado de programa desensamblado. Si el mando no lleva parámetros el desensamblado comienza en el punto en que haya terminado el anterior (si no ha habido uno anterior comienza en cero). Si el mando lleva solo como parámetro la dirección inicial se desensamblan diez instrucciones a partir de dicha dirección. Si se dan las direcciones inicial y final con el mando de listado, éste comienza con la dirección inicial y termina cuando la dirección final quede contenida en una instrucción desensamblada.

EJEMPLOS

Listar el programa a partir de 1000₁₆

```
68fil: L 1000
<001000> 41F9 000C 3501          LEA      800001,A0
<001006> 117C 00A0 0014          MOVE.B  #$A0,20(A0)
<00100C> 612A                    BSR.S   2A (=001038)
<00100E> 117C 0064 0016          MOVE.B  #$64,22(A0)
<001014> 21FC 0000 00BE 0064     MOVE.L  #$000000BE,100
<00101C> 203C 0003 CFA2          MOVE.L  #$0003CFA2,D0
<001022> 01C8 0018              MOVEP.L D0,24(A0)
<001026> 117C 00A1 0014          MOVE.B  #$A1,20(A0)
<00102C> 117C 0018 0018          MOVE.B  #$18,24(A0)
<001032> 117C 0033 001A          MOVE.B  #$33,26(A0)
68fil:
```

NOMBRE

M - Modificar memoria

FORMATO

M d ["c"]

donde: d = dirección de memoria
c = cadena de caracteres (opcional)

DESCRIPCION

Permite modificar el contenido de la memoria a partir de una dirección. Si se incluye una cadena de caracteres como par metro, ésta se introduce en el lugar indicado. En el caso de no incluir una cadena de caracteres en los par metros, el monitor va mostrando repetidamente una word y permite su modificación mediante la introducción de un nuevo valor. Cuando se tecllea tan solo CR, el contenido de la word queda inalterado. Este proceso repetitivo se interrumpe con la tecla ESC.

EJEMPLOS

Introducir en la memoria a partir de la dirección 2000₁₆ el texto "FiloSoft"

```
68fil: M 2000 "FiloSoft"  
68fil:
```

Modificar el contenido de la memoria a partir de la dirección 2000₁₆

```
68fil: M 2000  
<002000> 0000 4669(cr)      //- nuevo valor 466916  
<002002> 7657 6C6F(cr)      //- nuevo valor 6C6F16  
<002004> 536F (cr)          // word inalterada  
<002006> 0000 6674(cr)      // nuevo valor 667416  
<002008> 6556 (esc)         // fin de la modificación  
68fil:
```

NOMBRE

P - Punto de ruptura

FORMATO

P [n [d]]

donde: n = número de punto de ruptura (1..7) (opcional)
d = dirección de memoria (solo si hay n) (opcional)

DESCRIPCION

Permite establecer puntos de ruptura en la ejecución de programas. Pueden establecerse hasta 7 puntos de ruptura, que se denotan con un número del 1 al 7. Cuando se especifique el mando completo (con número de punto de ruptura y dirección de memoria) el sistema acepta el nuevo punto. Si el número de punto de ruptura especificado estaba ya utilizado, un mensaje avisa de ello y éste no es alterado. Si no se especifica una dirección el punto de ruptura indicado es eliminado en caso de existir. Por último, si sólo se indica 'P', se obtiene una lista de los puntos de ruptura establecidos.

EJEMPLOS

Listar los puntos de ruptura.

```
68fil: P
#5- 0046A6
68fil:
```

Establecer el punto de ruptura 2 en la dirección 2006₁₆.

```
68fil: P 2 2006
#2- 002006
#5- 0046A6
68fil:
```

Eliminar el punto de ruptura 2

```
68fil: P 2
#5- 0046A6
68fil:
```

NOMBRE

R - Ver/Modificar registros

FORMATO

R [r v]

donde: r = identificador de registro (opcional junto a v)

v = valor (opcional junto a r)

DESCRIPCION

Este mando permite ver y modificar el contenido de los registros del microprocesador. También muestra la instrucción que est siendo apuntada. Para ver los registros basta con indicar 'R', mientras que para modificar uno de ellos se añadir su nombre identificador y el nuevo valor a establecer. Los nombres identificadores de los registros son los establecidos por Motorola.

EJEMPLOS

Ver el contenido de los registros

```
68fil: R
D0=00000000 D1=00000000 D2=00000000 D3=00000000 | PC=00000000
D4=00000000 D5=00000000 D6=00000000 D7=00000000 |USP=0000F800
A0=00000000 A1=00000000 A2=00000000 A3=00000000 |SSP=0000FA00
A4=00000000 A5=00000000 A6=00000000 SR=0000 [...0.....]
<000000> 4669 6C6F NOT.W 6C6F(A1)
68fil:
```

Introducir el valor 2000₁₆ en el Program Counter (PC)

```
68fil: R PC 2000
D0=00000000 D1=00000000 D2=00000000 D3=00000000 | PC=00002000
D4=00000000 D5=00000000 D6=00000000 D7=00000000 |USP=0000F800
A0=00000000 A1=00000000 A2=00000000 A3=00000000 |SSP=0000FA00
A4=00000000 A5=00000000 A6=00000000 SR=0000 [...0.....]
<002000> 0000 0000 ORI.B #00,D0
68fil:
```

NOMBRE

S - Traza sin Saltos

FORMATO

S [d]

donde: d = dirección de memoria (opcional)

DESCRIPCION

Mediante este mando se ejecuta el programa desde la dirección apuntada por el PC hasta la siguiente instrucción en la memoria. Si se da una dirección como par metro, se carga el PC con dicha dirección antes de ejecutar. Al terminar la ejecución se muestran los registros, en los cuales se podrá ver el PC apuntando a la siguiente instrucción así como los efectos que haya podido tener la ejecución anterior sobre estos. Este mando equivale a una ejecución normal con un punto de ruptura en la instrucción siguiente.

EJEMPLOS

Ejecutar paso a paso a partir de la dirección 2000_{16} de forma que la subrutina que se halla en $202A_{16}$ se ejecute en un solo paso.

```
68fil: S 2000
D0=000000FF D1=00000000 D2=00000000 D3=00000000 | PC=00002004
D4=00000000 D5=00000000 D6=00000000 D7=00000000 |USP=0000F800
A0=00000000 A1=00000000 A2=00000000 A3=00000000 |SSP=0000FA00
A4=00000000 A5=00000000 A6=00000000 SR=0000 [...0.....]
<002004> 6106 BSR.S 202A
68fil: S
D0=000000FF D1=000000FF D2=00000000 D3=00000000 | PC=00002006
D4=00000000 D5=00000000 D6=00000000 D7=00000000 |USP=0000F800
A0=00000000 A1=00000000 A2=00000000 A3=00000000 |SSP=0000FA00
A4=00000000 A5=00000000 A6=00000000 SR=0004 [...0..Z..]
<002006> 20C0 MOVE.L D0,(A0)+
68fil:
```

NOMBRE

T - Traza

FORMATO

T [d]

donde: d = dirección de memoria (opcional)

DESCRIPCION

Mediante este mando se ejecuta una sola instrucción que está direccionada por el PC. Si se da una dirección como par metro, se carga el PC con dicha dirección antes de ejecutar la instrucción. Al terminar la ejecución se muestran los registros, en los cuales se podrá ver el PC apuntando a la siguiente instrucción así como los efectos que haya podido tener la inmediatamente ejecutada sobre estos.

EJEMPLOS

Ejecutar paso a paso a partir de la dirección 2000₁₆.

```

68fil: T 2000
D0=000000FF D1=00000000 D2=00000000 D3=00000000 | PC=00002004
D4=00000000 D5=00000000 D6=00000000 D7=00000000 |USP=0000F800
A0=00000000 A1=00000000 A2=00000000 A3=00000000 |SSP=0000FA00
A4=00000000 A5=00000000 A6=00000000 SR=0000 [...0.....]
<002004> 6106 BSR.S 202A
68fil: T
D0=000000FF D1=00000000 D2=00000000 D3=00000000 | PC=0000202A
D4=00000000 D5=00000000 D6=00000000 D7=00000000 |USP=0000F800
A0=00000000 A1=00000000 A2=00000000 A3=00000000 |SSP=0000FA00
A4=00000000 A5=00000000 A6=00000000 SR=0000 [...0.....]
<00202A> 4E56 FFF8 LINK A6,#F8
68fil:

```


NOMBRE

V - Volcar memoria

FORMATO

V [i [f]]

donde: i = dirección inicial de bloque (opcional)

f = dirección final de bloque (solo si hay i) (opcional)

DESCRIPCION

Muestra la memoria entre dos direcciones. En concreto se muestra la memoria desde de la dirección múltiplo de 16 ($XXXXX0_{16}$) más cercana por debajo de la dirección inicial y a lo largo de un numero entero de bloques de 16 bytes hasta sobrepasar la dirección final. Si no se especifica dirección final se muestran 256 bytes. Cuando no se indica ninguna dirección se lista a partir de la última dirección mostrada con anterioridad (si no se ha mostrado anteriormente se hace a partir de cero).

EJEMPLOS

Mostrar 256 bytes de memoria a partir de la dirección $FF1004_{16}$.

```
68fil: V FF1004
<FF1004> 02 63 6F 6D 70 6F 6E 65 6E 74 65 20 20 20 74 69 .componente...ti
<FF1014> 70 6F 20 20 20 20 20 70 6F 73 69 63 69 6F 6E 20 po.....posicion.
<FF1024> 20 20 65 73 71 75 65 6D 61 02 0D 0A 02 2D 2D 2D ..esquema....---
<FF1034> 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D -----
<FF1044> 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D -----
<FF1054> 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D -----
<FF1064> 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D -----
<FF1074> 2D 02 0D 0A 55 31 20 20 20 20 20 20 20 20 20 20 -...U1.....
<FF1084> 20 36 38 30 30 30 20 20 20 20 20 20 20 41 42 43 32 .68000.....ABC2
<FF1094> 33 20 20 20 31 0D 0A 55 32 20 20 20 20 20 20 20 20 3...1..U2.....
<FF10A4> 20 20 20 20 36 38 32 33 30 20 20 20 20 20 20 20 43 ....68230.....C
<FF10B4> 44 34 35 36 20 20 20 32 0D 0A 55 33 20 20 20 20 D456...2..U3....
<FF10C4> 20 20 20 20 20 20 20 36 38 36 38 31 20 20 20 20 .....68681....
<FF10D4> 20 20 42 43 35 36 20 20 20 20 33 0D 0A 55 34 20 ..BC56....3..U4.
<FF10E4> 20 20 20 20 20 20 20 20 20 20 20 20 31 34 38 38 20 20 .....1488..
<FF10F4> 20 20 20 20 20 42 36 20 20 20 20 20 20 33 0D 0A .....B6.....3..
68fil:
```

Mostrar la memoria entre las direcciones $FF1004_{16}$ y $FF1037_{16}$.

```
68fil: V FF1004 FF1037
<FF1004> 02 63 6F 6D 70 6F 6E 65 6E 74 65 20 20 20 74 69 .componente...ti
<FF1014> 70 6F 20 20 20 20 20 70 6F 73 69 63 69 6F 6E 20 po.....posicion.
<FF1024> 20 20 65 73 71 75 65 6D 61 02 0D 0A 02 2D 2D 2D ..esquema....---
<FF1034> 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D 2D -----
68fil:
```


X

Reinicializar el monitor

X

NOMBRE

X - Reinicializar el monitor

FORMATO

X

DESCRIPCION

Este mando es realiza un "reset" del monitor. De esta forma se evita tener que pulsar el botón que se encuentra en la placa para reinicializar el sistema.

El efecto observado por el usuario consiste en la puesta a cero de todos los registros y la situación en sus posiciones iniciales de los apuntadores de pilas (USP y SSP).

Puede utilizarse para forzar un test de memoria, para consultar el número de identificación de la placa, o para reinicializar los dispositivos 68230 y 68681.

EJEMPLOS

Reinicializar el sistema.

```
68fil: X
< aquí se verá realizarse el test de memoria y presentará>
< la pantalla inicial que consiste en el "menú de ayuda">
< y el número de identificación de la placa.>
68fil:
```


Sección 2: Formatos

68fil v2 Manual del software

INTRODUCCION

El monitor tiene la capacidad de tratar con cadenas de caracteres. Esto permite al usuario utilizar algunos parámetros de dicho tipo (ver B y M en la sección 1). Por otro lado el usuario dispone también de rutinas podrá utilizar en sus propios programas que ejecutan diversas acciones con cadenas (ver puts, gets, toupper, atoi, itoa, strcpy, strlen en la sección 4).

FORMATO

Una cadena consiste simplemente en una cadena de bytes distintos de cero finalizada con un byte cero.

Cuando se utilice una cadena como parámetro de un mando del monitor esta se especificará en código ASCII entre comillas dobles.

EJEMPLO

Una cadena dentro de un mando de monitor (escrita en ASCII entre dobles comillas):

```
68fil: B 0 FFFF "FiloSft"
```


INTRODUCCION

Un fichero conteniendo código S28 tiene el siguiente aspecto:

```
S214003000274C4120435552494F534944414420459B
S21400301053204C41204D41445245204445204C41CC
S20E003020204349454E4349412E2740
S804000000FB
```

como se puede notar el alfabeto utilizado por este código consiste tan solo en la letra 'S' y los dígitos y letras que son necesarios para especificar números en notación hexadecimal (0..9,A..F). Cada línea de código termina con los caracteres <cr> (retorno de carro, 0D₁₆) y <lf> (paso de línea, 0A₁₆) de forma que puede ser leído normalmente mediante un editor de texto.

ESTRUCTURA

Vamos a ver ahora cual la estructura de este código:

a	b	c	d	e	f
S	2	14	003000	274C4120435552494F53494441442045	9B
S	2	14	003010	53204C41204D41445245204445204C41	CC
S	2	0E	003020	204349454E4349412E27	40
S	8	04	000000		FB

Todas la líneas tienen la misma estructura, que es la formada por los campos que denominamos a, b, c, d, e, f. Cada campo contiene la siguiente información:

- Marca de comienzo 'S' (53₁₆), igual para todas la líneas.
- Indicador del tipo de línea. Puede tener dos valores:
 - 2 --> la línea contiene datos válidos.
 - 8 --> la línea es la última del bloque de código S28
- Número de bytes contenidos en los campos d, e y f.
- Dirección de memoria en la cual se deberán almacenar los datos.
- Datos.
- Complemento a 1 de la suma (en un byte) de los valores contenidos en los campos c, d y e. Se utiliza como método de comprobación por el receptor de que la transmisión ha sido correcta.

En los campos c, e y f se interpreta cada par de caracteres como un byte en notación hexadecimal.

EJEMPLO

Analizaremos ahora una línea de código:

a	b	c	d	e	f
S	2	14	003000	274C4120435552494F53494441442045	9B

- a) Carácter 'S' de comienzo de la línea.
 b) Carácter '2' que indica que la línea contiene datos.
 c) Número hexadecimal 14_{16} , que indica el número de bytes de los campos d, e y f es de 20 ($3+16+1$).
 d) Dirección $3000H$ de la memoria del microprocesador en la que deben almacenarse los datos.
 e) Datos: 27_{16} , $4C_{16}$, 41_{16} , 20_{16} , 43_{16} , 55_{16} , 52_{16} , 49_{16} , $4F_{16}$, 53_{16} , 49_{16} , 44_{16} , 41_{16} , 44_{16} , 20_{16} , 45_{16}
 f) Suma de los bytes de los campos c, d y e complementada a 1.

campo	valor	suma acumulada
-----	-----	-----
c	14_{16}	14_{16}
d	00_{16}	14_{16}
d	30_{16}	44_{16}
d	00_{16}	44_{16}
e	27_{16}	$6B_{16}$
e	$4C_{16}$	$B7_{16}$
e	41_{16}	$F8_{16}$
e	20_{16}	$1-18_{16}$
e	43_{16}	$1-5B_{16}$
e	55_{16}	$1-B0_{16}$
e	52_{16}	$1-02_{16}$
e	49_{16}	$1-4B_{16}$
e	$4F_{16}$	$1-9A_{16}$
e	53_{16}	$1-ED_{16}$
e	49_{16}	$2-36_{16}$
e	44_{16}	$2-7A_{16}$
e	41_{16}	$2-BB_{16}$
e	44_{16}	$2-FF_{16}$
e	20_{16}	$3-1F_{16}$
e	45_{16}	$3-64_{16}$

$64_{16} = 0110\ 0100 \rightarrow$ complemento a 1 $\rightarrow 1001\ 1011 = 9B_{16}$

Sección 3: SERMONES

68fil v2 Manual del Software

El usuario del entrenador 68fil dispone de una serie de rutinas básicas que van orientadas principalmente a permitirle la introducción de datos a partir de su teclado y la presentación de resultados en la pantalla. Estas rutinas manejan tanto datos numéricos como cadenas de caracteres (ver cadenas en la sección 2). Por tratarse de rutinas de servicio que proporciona el monitor se han denominado SERMONES (SERvicios del MONitor).

El programador dispone de total libertad para el uso de recursos (registros, periféricos, etc.) con la única restricción de que sus programas deberán acabar con el SERMON FPROG (ver en esta misma sección) que marca el punto en el cual debe volver a tomar el control el monitor.

El monitor utiliza la puerta serie A para comunicarse con el terminal, no obstante el programa de usuario puede cambiar la programación de dicha puerta ya que cuando el monitor vuelve a tomar el control (mediante un FPROG) se encarga de reconfigurarla automáticamente. Naturalmente en caso de que el usuario utilice la puerta serie A solo podrá ejecutar su programa de manera continua. No podrá utilizar las facilidades de test del monitor (trazas, puntos de ruptura, etc.)

NOMBRE

Fprog - Marca el final de un programa de usuario.

DESCRIPCION

Marca el final de un programa de usuario.

LLAMADA

TRAP #15
DC.W 0

SINOPSIS**RELACION CON OTRAS RUTINAS**

NOMBRE

Getch - Toma un carácter del canal serie.

DESCRIPCION

Toma un carácter del canal serie A. No utiliza ningún tipo de filtro. Es bloqueante, es decir, si no se recibe ningún carácter por el canal queda a la espera.

LLAMADA

TRAP #15
DC.W 1

SINOPSIS

Salida: D7(B) carácter leído.
Modificación: CCR.

RELACION CON OTRAS RUTINAS

Inversa: putch

NOMBRE

Putch - pone un carácter en el canal serie.

DESCRIPCION

Pone un carácter en el canal serie A. Puede bloquear por recepción de un XOFF (11H) hasta la llegada de un XON (13H). También se puede bloquear si se realiza protocolo de comunicaciones por hardware.

LLAMADA

TRAP #15
DC.W 2

SINOPSIS

Entrada: D7(B) carácter a enviar.
Modificación: CCR

RELACION CON OTRAS RUTINAS

Inversa: getch.

NOMBRE

gets - toma una cadena de caracteres del canal serie.

DESCRIPCION

Toma una cadena de caracteres del canal serie A (ver Cadena en sección 2-Formatos). Da por terminada la cadena al recibir un carácter CR (retorno de carro, 0D₁₆) o un ESC (escape, 1B₁₆). Es bloqueante, es decir, no se devuelve el control hasta que se recibe un carácter finalizador de la cadena. Filtra los caracteres de control. Hace eco de los caracteres no filtrados por el mismo canal. El carácter 'backspace' (08₁₆) tiene como eco la serie 'backspace'+espacio+'backspace' (08₁₆, 20₁₆, 08₁₆).

LLAMADA

TRAP #15
DC.W 3

SINOPSIS

Entrada: A0(L) dirección de memoria donde se almacenará la cadena recibida.

Salida: EQ - terminado por 'retorno de carro'

NE - terminado por 'escape'

Modificación: CCR

RELACION CON OTRAS RUTINAS

Inversa: puts.

Utiliza: getch, putch.

NOMBRE

Puts - Pone una cadena de caracteres en el canal serie.

DESCRIPCION

Pone una cadena de caracteres en el canal serie A (ver Cadena en sección 2-Formatos). Puede bloquear por recepción de un XOFF (11H) hasta la llegada de un XON (13H). Tambi,n se puede bloquear si se realiza protocolo de comunicaciones por hardware.

LLAMADA

TRAP #15
DC.W 4

SINOPSIS

Entrada: A0(L) apuntador a la cadena.
Modificación: CCR

RELACION CON OTRAS RUTINAS

Inversa: gets.
Utiliza: putch.

NOMBRE

Toupper - convierte minúsculas en mayúsculas en una cadena.

DESCRIPCION

En una cadena cambia todos los bytes correspondientes en código ASCII a las letras minúsculas por los correspondientes a sus mayúsculas (ver Cadena en sección 2-Formatos). Los demás caracteres quedan inalterados.

LLAMADA

TRAP #15
DC.W 5

SINOPSIS

Entrada: A0(L) apuntador a la cadena.
Modificación: CCR

RELACION CON OTRAS RUTINAS

NOMBRE

Atoi - convierte una cadena en un número entero.

DESCRIPCION

Dada una cadena de caracteres que contiene un número entero expresado en ASCII devuelve en un registro dicho número (ver Cadena y Número en sección 2-Formatos). En caso de que la cadena no contenga un número el resultado es indeterminado, quedando esto reflejado en el CCR(ver sinopsis).

LLAMADA

TRAP #15

DC.W 6

SINOPSIS

Entrada: A0(L) apuntador a la cadena.

Salida: EQ - ejecución correcta.

D7(L) resultado entero.

A0(L) apuntando justo detrás de la cadena de entrada.

NE - ejecución incorrecta.

D7(L) indeterminado.

A0(L) apuntando al carácter siguiente al no aceptado

Modificación: CCR

RELACION CON OTRAS RUTINAS

Inversa: Itoa.

Utiliza: Toupper.

NOMBRE

Itoa - convierte un número entero en una cadena.

DESCRIPCION

Dado un número entero construye una representación hexadecimal del mismo en una cadena de caracteres ASCII (ver Cadena en sección 2-Formatos).

LLAMADA

TRAP #15
DC.W 7

SINOPSIS

Entrada: A0(L) Apuntador al espacio para la cadena.
D7(L) Número entero.

Salida: A0(L) Apuntando al byte siguiente al fin de la cadena.

Modificación: CCR

RELACION CON OTRAS RUTINAS

Inversa: atoi.

NOMBRE

Strcpy - Copia una cadena en otra.

DESCRIPCION

Dada una cadena de caracteres y una posición de memoria se sitúa a partir de ella una copia de la cadena.

LLAMADA

TRAP #15
DC.W 8

SINOPSIS

Entrada: A1(L) apuntador a la cadena origen.

A0(L) apuntador a la cadena destino.

Salida: A0(L) apunta al byte siguiente al final de la cadena destino.

Modificación: CCR

RELACION CON OTRAS RUTINAS

NOMBRE

Strlen - mide la longitud de una cadena en bytes.

DESCRIPCION

Dada una cadena de caracteres devuelve el número de bytes que la forman sin contar el cero que indica el fin de la cadena.

LLAMADA

TRAP #15
DC.W 9

SINOPSIS

Entrada: A0(L) apuntando a la cadena a medir.
Salida: A0(L) apuntando al byte siguiente al final de la cadena.
D7(W) longitud de la cadena.
Modificación: CCR

RELACION CON OTRAS RUTINAS

NOMBRE

Memcpy - copia memoria.

DESCRIPCION

Copia un bloque de memoria en una nueva posición. La copia siempre se hace de forma que, si hay solapamiento entre los bloques origen y destino, el bloque destino quede como el original (nótese que si hay solapamiento el bloque origen queda modificado).

LLAMADA

TRAP #15
DC.W 10

SINOPSIS

Entrada: A0(L) apuntador al bloque a copiar
A1(L) apuntador a la posición destino
D7(W) numero de bytes a copiar
Modificación: CCR, A0(L), A1(L), D7(W).

RELACION CON OTRAS RUTINAS

NOMBRE

Linputs - Pone en el canal serie un paso de línea y una cadena.

DESCRIPCION

Envía al canal serie A los caracteres $0D_{16}$ (CR) y $0A_{16}$ (LF) seguidos de una cadena de caracteres (ver Cadena en sección 2-Formatos). Puede bloquear por recepción de un XOFF (11_{16}) hasta la llegada de un XON (13_{16}). También se puede bloquear si se realiza protocolo de comunicaciones por hardware.

LLAMADA

TRAP #15
DC.W 11

SINOPSIS

Entrada: A0(L) apuntador a la cadena.
Modificación: CCR

RELACION CON OTRAS RUTINAS

Utiliza: putch.
Otras relaciones: gets, puts.

NOMBRE

Kbhit - Consulta si se ha pulsado una tecla.

DESCRIPCION

Consulta el canal serie A para ver si se encuentra esperando en el periférico un carácter recibido del teclado.

LLAMADA

TRAP #15
DC.W 12

SINOPSIS

Salida: EQ - No se ha pulsado una tecla.
NE - Se pulsado una tecla.
Modificación: CCR

RELACION CON OTRAS RUTINAS